

Estimating Attraction Basin Sizes

Leticia Hernando¹, Alexander Mendiburu¹, and Jose A. Lozano^{1,2}

¹ Intelligent Systems Group,
University of the Basque Country (UPV/EHU), 20018 Donostia, Spain
{leticia.hernando, alexander.mendiburu, ja.lozano}@ehu.eus
² Basque Center for Applied Mathematics (BCAM), 48009 Bilbao, Spain

Abstract. The performance of local search algorithms is influenced by the properties that the neighborhood imposes on the search space. Among these properties, the number of local optima has been traditionally considered as a complexity measure of the instance, and different methods for its estimation have been developed. The accuracy of these estimators depends on properties such as the relative attraction basin sizes. As calculating the exact attraction basin sizes becomes unaffordable for moderate problem sizes, their estimations are required. The lack of techniques achieving this purpose encourages us to propose two methods that estimate the attraction basin size of a given local optimum. The first method takes uniformly at random solutions from the whole search space, while the second one takes into account the structure defined by the neighborhood. They are tested on different instances of problems in the permutation space, considering the swap and the adjacent swap neighborhoods.

1 Introduction

Local search algorithms have been proved as efficient methods for solving hard permutation-based COPs. These methods rely on a neighborhood structure over the search space. The properties of this neighborhood can cause dramatic differences in the performance of those local search methods [9, 10, 12]. Thus, the same local search algorithm can produce different results in the same instance depending on the neighborhood chosen, because different neighborhoods draw different shapes (ruggedness) in the landscapes.

One of the features imposed by the neighborhood is the number of local optima. This number of local optima has commonly been taken as a complexity measure of an instance when solving it with a local search algorithm, and many authors have tried to estimate it [1–6]. One of the results found when developing these techniques for predicting the number of local optima was that their accuracy is highly affected by the variance of the attraction basin sizes of the local optima. In general, the more uniform the attraction basin sizes are, the better the prediction. However, there are methods that are able to provide good estimations for instances where the variance of the attraction basin sizes is extremely large [6]. Looking at these methods, we can observe that their estimations rely on the concept of sample coverage, that is, the sum of the relative sizes

of the attraction basins of the local optima observed in the sample. Inspired by this finding, our interest lies in finding methods that calculate the attraction basin size of any given local optimum. Unfortunately, in the literature there is a lack of studies fulfilling this aim.

The most intuitive way of obtaining the exact attraction basin of a local optimum would be by exhaustively applying the local search algorithm starting from each solution of the search space, and taking those solutions that finish at such local optimum. Of course, this is useless because if we were able to perform this process, we would be able to solve the optimization problem. Hence, another method for calculating the attraction basin of a local optimum is by considering, as the starting point, this local optimum. Then, applying a recursive procedure that checks at each time whether the neighbors of the current solution belong to the attraction basin. This procedure finishes when there are no more possible solutions to add to the attraction basin. In this last method we do not need to evaluate all the solutions of the search space. However, in general, the number of solutions to evaluate grows exponentially with respect to the problem size. Therefore, for large problem sizes, there could be local optima for which it becomes computationally intractable to exactly calculate their attraction basins.

The fact that there is no known method that calculates, in polynomial time, the exact attraction basins of the local optima, or at least, their sizes, leads us to focus on methods that estimate the attraction basin sizes. An easy and simple method for estimating the sizes of the attraction basins consists of applying a local search to a sample of solutions, estimating each proportion of the size of the attraction basin of the local optima as the proportion of times that it has been reached in the sample [7, 11]. However, this method has a major weakness: it is supposed that there are no more local optima in the search space except just those encountered in the sample. Of course, in general, this is not true.

Given a local optimum, we propose in this paper two methods for estimating its attraction basin size. Both methods start from the local optimum for which we want to estimate the attraction basin size. The first method consists of taking solutions uniformly at random from the whole search space. In the second method, the search space is divided in different subsets, which correspond to the sets of permutations at different distances. Three different sample strategies are used to sample these subsets. The way of choosing the sample sizes for the different chunks of the search space could help in the estimation, as well as could disorientate it, because, once chosen the sample strategy, the performance of this proposal will depend on the landscape properties. Considering the adjacent swap and swap neighborhoods, we test the methods on instances of two permutation-based combinatorial optimization problems: Permutation Flowshop Scheduling Problem (PFSP) and Linear Ordering Problem (LOP).

The rest of the paper is organized as follows. In Section 2, we explain in detail both methods and, in Section 3, we compare their accuracy when they are applied to the instances using the different neighborhoods. Finally, in Section 4, we review the main conclusions obtained.

2 Methods for Estimating the Attraction Basin Sizes

A **neighborhood** \mathcal{N} in a search space Ω is a mapping that assigns to each solution $\pi \in \Omega$ a set of neighboring solutions $\mathcal{N}(\pi)$. Particularly, we work with instances where Ω is the space of permutations of size n , so $|\Omega| = n!$. Two examples of neighborhoods in the space of permutations are the adjacent swap and the swap operators. The **adjacent swap neighborhood** considers two neighboring solutions if one is generated by swapping two adjacent elements of the other. While two solutions are neighbors under the **swap neighborhood** if one is the result of swapping any two elements of the other, not necessarily adjacent.

Supposing a minimization problem, a solution $\pi^* \in \Omega$ is a **local optimum** if $f(\pi^*) \leq f(\pi)$, $\forall \pi \in \mathcal{N}(\pi^*)$. Each local optimum π^* has associated its attraction basin $\mathcal{B}(\pi^*)$, that is, the set composed of all the solutions that, after applying a local search algorithm starting with those solutions, finishes in π^* . Particularly, we use a deterministic best-improvement local search (the steps followed by this algorithm are specified in Algorithm 1). Denoting by \mathcal{H} the operator that associates to each solution π the local optimum π^* obtained after applying the algorithm, we can formally define the **attraction basin of a local optimum** as: $\mathcal{B}(\pi^*) = \{\pi \in \Omega \mid \mathcal{H}(\pi) = \pi^*\}$. Notice that the neighbors are evaluated in a specific order, so that, in the case of two neighbors having the same function value, the algorithm will always choose that which was encountered first (the neighbor that is the result of swapping the smallest items $\pi(i)$ and $\pi(j)$).

We propose two estimators to calculate the size of the attraction basin of a given local optimum π^* . This estimation will be denoted by $|\hat{\mathcal{B}}(\pi^*)|$.

Algorithm 1 Deterministic best-improvement local search algorithm

```

1: Choose an initial solution  $\pi \in \Omega$ 
2: repeat
3:    $\pi^* = \pi$ 
4:   for  $i = 1 \rightarrow |\mathcal{N}(\pi^*)|$  do
5:     Choose  $\sigma_i \in \mathcal{N}(\pi^*)$ 
6:     if  $f(\sigma_i) < f(\pi)$  then
7:        $\pi = \sigma_i$ 
8:     end if
9:   end for
10: until  $\pi = \pi^*$ 

```

2.1 Uniformly at Random Method (UM)

Given a local optimum π^* , we sample solutions uniformly at random from the search space counting those that belong to its attraction basin. That is, we take a sample of size M of random initial solutions: $S = \{\pi_1, \pi_2, \dots, \pi_M\} \subseteq \Omega$. The number of those solutions that belong to the attraction basin of π^* divided by

the total number of solutions evaluated (M) is the estimated proportion of the attraction basin size of π^* over the size of the search space $|\Omega|$. So, the attraction basin size of π^* is this proportion multiplied by the size of the search space ($n!$). In Algorithm 2, we specify these steps to follow.

Algorithm 2 Uniformly at random Method (UM) to estimate the size of the attraction basin of a local optimum π^*

```

1: Input:  $M$ 
2: Initialize  $InAB = 0$ 
3: for  $i = 1 \rightarrow M$  do
4:   take a random permutation  $\pi_i \in \Omega$ 
5:    $\sigma = \mathcal{H}(\pi_i)$ 
6:   if  $\sigma == \pi^*$  then
7:      $InAB++$ 
8:   end if
9: end for
10:  $|\hat{\mathcal{B}}(\pi^*)| = \frac{InAB}{M} \cdot n!$ 
11: Output:  $|\hat{\mathcal{B}}(\pi^*)|$ 

```

2.2 Distance-based Method (DM)

In this second proposal, we do not take a random sample directly from the whole search space Ω . Instead, given a local optimum π^* , we choose the solutions from different subsets of Ω related to π^* . That is, we consider the different subsets $D_i = \{\pi_1^i, \pi_2^i, \dots, \pi_{|D_i|}^i\} \subseteq \Omega$ that are composed of those solutions at distance i from the local optimum π^* . We say that two permutations π_1 and π_2 are at distance i if, starting from π_1 , and moving from neighboring to neighboring solutions, the length of the smallest path until reaching π_2 is i . Particularly, two neighboring permutations are at distance one.

Notice that any permutation in $\Omega \setminus \{\pi^*\}$ should belong to one, and just one, of these subsets D_i . That is:

- (i) $D_i \cap D_j = \emptyset, \forall i \neq j$.
- (ii) $\bigcup_i D_i \cup \{\pi^*\} = \Omega$.

So, given the local optimum π^* , we take samples S_1, S_2, \dots , of uniformly at random solutions at distances $1, 2, \dots$, respectively, from π^* :

$$S_1 = \{\pi_1^1, \pi_2^1, \dots, \pi_{M_1}^1\} \subseteq D_1; S_2 = \{\pi_1^2, \pi_2^2, \dots, \pi_{M_2}^2\} \subseteq D_2; \dots$$

We use the methods described in [8] to obtain these uniformly at random solutions π_j^i for the different distances. In order to estimate the attraction basin size of π^* , we proceed in a similar way to the previous method but, we work with the different subsets D_i independently. That is, we record the number of

solutions that belong to the attraction basin of π^* in each sample set S_i , divided by the sample size considered for each distance M_i , and multiplied by the total number of permutations that exist in each subset D_i . Therefore, the sum of these quantities obtained for each distance plus one (π^* itself is in its attraction basin and has not been considered in any subset) is the resultant attraction basin size of the local optimum π^* . This process is detailed in Algorithm 3, where $MaxDist$ denotes the maximum distance between two permutations and $|D_{dist}|$ refers to the number of permutations at distance $dist$. Both the maximum distance and the number of permutations at a given distance depend on the problem size and the neighborhood used. The input parameter of the algorithm is $\mathbf{M} = \{M_1, \dots, M_{MaxDist}\}$, that is, we need to set in advance the sample size used at each distance $dist$. On the one hand, different subsets D_i have different sizes, which could lead us to change the sample size taking values proportional to $|D_i|$. On the other hand, the closer a permutation to the optimum, the more probable it belongs to its attraction basin. Thus, we could consider the possibility of taking more random solutions in the subsets D_i than D_j , with $i < j$, or even, to stop taking solutions from a certain distance on.

Algorithm 3 Distance-based Method (DM) to estimate the size of the attraction basin of a local optimum π^* .

```

1: Input:  $\mathbf{M} = \{M_1, \dots, M_{MaxDist}\}$ 
2:  $|\hat{\mathcal{B}}(\pi^*)| = 1$ 
3: for  $dist = 1 \rightarrow MaxDist$  do
4:   Initialize  $InAB = 0$ 
5:   for  $j = 1 \rightarrow M_{dist}$  do
6:     take a random permutation  $\sigma \in D_{dist}$ 
7:      $\pi = \mathcal{H}(\sigma)$ 
8:     if  $\pi == \pi^*$  then
9:        $InAB++$ 
10:    end if
11:  end for
12:   $|\hat{\mathcal{B}}(\pi^*)| = |\hat{\mathcal{B}}(\pi^*)| + \frac{InAB}{M_{dist}} \cdot |D_{dist}|$ 
13: end for
14: Output:  $|\hat{\mathcal{B}}(\pi^*)|$ 

```

3 Experiments

We analyze and compare the two proposed methods for estimating the sizes of the attraction basins of the local optima for instances of different problems and considering different neighborhoods. We work with instances of the PFSP and the LOP, and we focus on the adjacent swap and swap neighborhoods. For the PFSP we consider 5 instances with 10 jobs and 5 machines, obtained from the well-known benchmark proposed by Taillard. The 5 instances of the LOP have

been obtained from the xLOLIB benchmark, and the matrix size considered is 10×10 . So, in both problems the size of the search space is $10!$. Notice that the problem size is quite small. The reason is that, in order to measure the accuracy of the methods, we calculate the exact attraction basin size of each of the local optima of the instances, and thus, working with large permutation sizes is computationally unaffordable. Regarding the parameters of the algorithms, we specify the sample sizes used. For the first method (UM) we choose samples of sizes: $M = \{1125, 2250, 4500\}$. For the second method (DM), we need to fix different sample sizes M_i according to the distance. In order to study different possibilities, for the second method we consider three different cases:

1. **Equal Sample sizes for each distance (ES):** $M_i = M_j, \forall i \neq j$.
2. **Sample sizes Proportional to the number of permutations at each distance (SP):** $M_i \propto |D_i|$.
3. **Sample sizes Decreasing as the distance increases (SD):** $M_i \propto \frac{1}{i}$, and $M_i = 0, i > MaxDist/2$.

We should use the same (or almost similar) total sample size, when comparing the second method (DM) with the first one (UM). So, we need to choose, in this case, M_i such that

$$M \approx \sum_{i=1}^{MaxDist} M_i \approx \begin{cases} 1125 \\ 2250 \\ 4500 \end{cases} \quad (1)$$

We show in Table 1 the sample sizes used at each distance according to the neighborhood, in order to fulfill equation (1). In Table 2, the number of solutions of size 10 at each distance from a given permutation for the adjacent swap and swap neighborhoods is facilitated. Both algorithms UM and DM are applied 10 times to each local optimum for each sample size, and the average estimations of the attraction basin sizes are recorded. From now on, instead of considering just two methods, we will refer to four: UM, DM-ES, DM-SP and DM-SD.

Table 1. Sample sizes used in the methods.

UM	ADJACENT SWAP			SWAP		
	DM-ES	DM-SP	DM-SD ($i \leq 22$)	DM-ES	DM-SP	DM-SD ($i \leq 4$)
1125	25	$\lceil \frac{ D_i }{3300} \rceil$	$\lceil \frac{302}{i} \rceil$	125	$\lceil \frac{ D_i }{3240} \rceil$	$\lceil \frac{540}{i} \rceil$
2250	50	$\lceil \frac{ D_i }{1630} \rceil$	$\lceil \frac{608}{i} \rceil$	250	$\lceil \frac{ D_i }{1616} \rceil$	$\lceil \frac{1080}{i} \rceil$
4500	100	$\lceil \frac{ D_i }{811} \rceil$	$\lceil \frac{1217}{i} \rceil$	500	$\lceil \frac{ D_i }{807} \rceil$	$\lceil \frac{2160}{i} \rceil$

In Table 3, the average relative error (in absolute value), $\frac{||\hat{\mathcal{B}}(\pi^*)| - |\mathcal{B}(\pi^*)||}{|\mathcal{B}(\pi^*)|}$, and the variance (in brackets) given by each method for all the local optima of each instance from the 10 repetitions, are reported, for the UM, the DM-ES, the DM-SP and the DM-SD, and according to the problem and neighborhood considered. As we can appreciate, when estimating the attraction basin sizes, there

Table 2. Number of permutations of size 10 at different distances from a given solution.

ADJACENT SWAP										SWAP	
dist	#perms	dist	#perms	dist	#perms	dist	#perms	dist	#perms	dist	#perms
1	9	10	21670	19	211089	28	162337	37	8095	1	45
2	44	11	32683	20	230131	29	135853	38	4489	2	870
3	155	12	47043	21	243694	30	110010	39	2298	3	9450
4	440	13	64889	22	250749	31	86054	40	1068	4	63273
5	1068	14	86054	23	250749	32	64889	41	440	5	269325
6	2298	15	110010	24	243694	33	47043	42	155	6	723680
7	4489	16	135853	25	230131	34	32683	43	44	7	1172700
8	8095	17	162337	26	211089	35	21670	44	9	8	1026576
9	13640	18	187959	27	187959	36	13640	45	1	9	362880

is not a best overall method. We observe that, clearly, on average terms, the method that provides the best results for the instances of the PFSP considering the adjacent swap neighborhood is DM-SD, where we find the lowest errors and very small variances. For the instances of the LOP considering the adjacent swap neighborhood, the DM-SD performs well, but we also find good results for the DM-ES. For both problems, when using the swap neighborhood, the results given by DM-SD are really bad (high errors and variances). This is because the DM-SD takes a lower number of solutions as the distance to the local optima increases, and it stops taking solutions at $MaxDist/2$. The bad performance of this method indicates that, when using the swap neighborhood, the attraction basins of the local optima are composed by a high number of solutions that are far from them, but this does not happen when using the adjacent swap. Of course, even if we increase the sample size with this method, as we stop considering solutions at certain distance, for both neighborhoods we do not obtain better results. The best method on average for almost all instances for the swap is DM-ES. So, for this neighborhood it seems convenient to take the same number of solutions at different distances. As a general rule, the higher the sample size the lower the average relative errors and variances. Except for DM-SD with the swap neighborhood, that the results are almost similar for all sample sizes.

We carry out a statistical analysis to compare the estimations obtained for the different methods. A nonparametric Friedman's test with level of significance $\alpha = 0.05$ is used to test if there are statistical significant differences between the estimations provided by the 4 methods in the different scenarios (according to problem and neighborhood). As we find statistical differences in all the cases, we proceed with a post-hoc test which carries out all pairwise comparisons. Particularly, we use the Holm's procedure fixing the level of significance to $\alpha = 0.05$. In the case of the PFSP with the adjacent swap neighborhood, we find that the best method is the DM-SD with significant differences. When using the swap neighborhood, the best method is the DM-ES but with no significant differences with the DM-SP. For the LOP and the adjacent swap, the best methods are the DM-SD and the DM-ES with no significant differences, while for the swap the UM, the DM-ES and the DM-SP are the best methods without significant differences among them. Of course, for both problems using the swap, the worst performing one is the DM-SD with significant differences.

Table 3. Average relative errors (and variances) of the attraction basin sizes of the local optima.

	$M = 1125$				$M = 2250$				$M = 4500$					
	UM	DM-ES	DM-SP	DM-SD	UM	DM-ES	DM-SP	DM-SD	UM	DM-ES	DM-SP	DM-SD		
PFSP	ADJACENT S.	1	1.16 (5.13)	0.19 (0.03)	0.56 (0.33)	0.16 (0.03)	0.90 (1.77)	0.14 (0.02)	0.49 (0.25)	0.12 (0.02)	0.75 (1.08)	0.10 (0.01)	0.40 (0.14)	0.09 (0.01)
		2	1.30 (4.61)	0.15 (0.02)	0.55 (0.30)	0.12 (0.01)	1.06 (2.16)	0.11 (0.01)	0.50 (0.22)	0.09 (0.01)	0.84 (1.06)	0.11 (0.01)	0.43 (0.14)	0.07 (0.00)
		3	1.33 (5.26)	0.14 (0.01)	0.51 (0.25)	0.10 (0.01)	1.11 (2.95)	0.10 (0.01)	0.47 (0.21)	0.08 (0.00)	0.83 (1.27)	0.08 (0.00)	0.42 (0.14)	0.07 (0.00)
		4	1.49 (8.77)	0.14 (0.02)	0.54 (0.31)	0.11 (0.01)	1.17 (3.05)	0.11 (0.01)	0.51 (0.24)	0.08 (0.01)	1.02 (2.27)	0.12 (0.01)	0.45 (0.15)	0.07 (0.00)
		5	0.60 (0.46)	0.24 (0.05)	0.46 (0.22)	0.23 (0.04)	0.45 (0.24)	0.18 (0.03)	0.35 (0.13)	0.17 (0.03)	0.34 (0.15)	0.13 (0.01)	0.26 (0.07)	0.13 (0.01)
SWAP		1	0.08 (0.01)	0.05 (0.00)	0.07 (0.00)	0.43 (0.05)	0.06 (0.01)	0.04 (0.00)	0.05 (0.01)	0.43 (0.05)	0.03 (0.00)	0.03 (0.00)	0.04 (0.00)	0.42 (0.05)
		2	0.30 (0.13)	0.13 (0.02)	0.25 (0.10)	0.24 (0.03)	0.24 (0.11)	0.09 (0.01)	0.20 (0.09)	0.21 (0.02)	0.19 (0.06)	0.07 (0.00)	0.14 (0.03)	0.21 (0.02)
		3	0.22 (0.10)	0.10 (0.01)	0.17 (0.03)	0.24 (0.03)	0.15 (0.05)	0.08 (0.00)	0.13 (0.02)	0.25 (0.03)	0.10 (0.01)	0.05 (0.00)	0.10 (0.02)	0.23 (0.03)
		4	0.40 (0.52)	0.13 (0.01)	0.29 (0.11)	0.21 (0.02)	0.22 (0.12)	0.10 (0.01)	0.20 (0.06)	0.18 (0.02)	0.19 (0.07)	0.07 (0.00)	0.15 (0.04)	0.17 (0.02)
		5	0.04 (0.00)	0.05 (0.00)	0.05 (0.00)	0.55 (0.05)	0.03 (0.00)	0.03 (0.00)	0.02 (0.00)	0.54 (0.05)	0.02 (0.00)	0.03 (0.00)	0.02 (0.00)	0.55 (0.05)
LOP	ADJACENT S.	1	0.93 (2.51)	0.21 (0.03)	0.56 (0.44)	0.18 (0.03)	0.79 (1.81)	0.15 (0.02)	0.48 (0.25)	0.13 (0.01)	0.55 (0.77)	0.11 (0.01)	0.37 (0.14)	0.10 (0.01)
		2	1.37 (10.99)	0.20 (0.03)	0.65 (0.48)	0.16 (0.02)	1.02 (2.29)	0.15 (0.02)	0.57 (0.29)	0.12 (0.01)	0.79 (1.36)	0.11 (0.01)	0.48 (0.22)	0.10 (0.01)
		3	1.64 (28.06)	0.24 (0.05)	0.73 (0.80)	0.20 (0.04)	1.50 (16.84)	0.18 (0.03)	0.68 (0.57)	0.16 (0.02)	1.26 (6.72)	0.15 (0.01)	0.61 (0.36)	0.14 (0.01)
		4	0.44 (0.19)	0.26 (0.09)	0.36 (0.15)	0.27 (0.07)	0.34 (0.21)	0.20 (0.04)	0.27 (0.12)	0.20 (0.04)	0.26 (0.11)	0.13 (0.01)	0.21 (0.08)	0.13 (0.01)
		5	0.33 (0.13)	0.23 (0.05)	0.26 (0.10)	0.27 (0.07)	0.19 (0.05)	0.17 (0.04)	0.19 (0.04)	0.19 (0.03)	0.14 (0.02)	0.11 (0.01)	0.14 (0.03)	0.13 (0.02)
SWAP		1	0.18 (0.03)	0.10 (0.01)	0.15 (0.03)	0.33 (0.07)	0.13 (0.03)	0.09 (0.01)	0.12 (0.02)	0.32 (0.07)	0.11 (0.01)	0.05 (0.00)	0.09 (0.01)	0.31 (0.07)
		2	0.32 (0.91)	0.12 (0.02)	0.23 (0.09)	0.33 (0.05)	0.16 (0.05)	0.08 (0.01)	0.18 (0.07)	0.30 (0.04)	0.13 (0.06)	0.06 (0.00)	0.12 (0.02)	0.29 (0.04)
		3	0.43 (0.27)	0.19 (0.03)	0.33 (0.12)	0.23 (0.02)	0.32 (0.13)	0.13 (0.01)	0.25 (0.07)	0.20 (0.02)	0.19 (0.04)	0.09 (0.01)	0.18 (0.04)	0.17 (0.02)
		4	0.02 (0.00)	0.03 (0.00)	0.05 (0.01)	0.60 (0.03)	0.04 (0.00)	0.04 (0.00)	0.04 (0.00)	0.60 (0.03)	0.02 (0.00)	0.02 (0.00)	0.01 (0.00)	0.59 (0.04)
		5	0.06 (0.00)	0.09 (0.01)	0.10 (0.02)	0.49 (0.04)	0.05 (0.00)	0.06 (0.00)	0.07 (0.01)	0.47 (0.05)	0.04 (0.00)	0.04 (0.00)	0.05 (0.01)	0.46 (0.05)

4 Conclusions

The estimation of the attraction basin sizes of the local optima helps in the understanding of the landscapes and is a useful information when analyzing the complexity of the instances for local search algorithms. For example, the knowledge about the attraction basin sizes becomes essential for the prediction of the number of local optima. On the one hand, knowing the relative attraction basin sizes facilitates the choice of a suitable method for estimating the number of local optima. On the other hand, if we have information about the proportion of the search space occupied by the attraction basins of those local optima observed in a sample, we would know the proportion of the search space that has not yet been explored. Consequently, this would be a valuable information to estimate the number of not seen local optima.

We present two methods for estimating the attraction basin size of a given local optimum. In the first method proposed (UM), the proportion of the attraction basin size of the local optimum is estimated as the proportion of solutions of a random sample that belong to it. The second method (DM), which is more computationally demanding, consists of taking random solutions at different distances and estimating the total size considering the sum of the estimations of the sizes that are related to each distance. In the DM, the sample size taken at each distance is of high relevance. We notice differences in considering different ways of choosing the samples for different neighborhoods. First, we consider the case where, for each distance, the same sample size is taken (DM-ES). Then, we take samples of sizes proportional to the number of permutations that are at the different distances (DM-SP). Finally, the samples are chosen with sizes that decrease as the distance to the local optimum increases, and we stop taking solutions further than $MaxDist/2$ (DM-SD).

The main result observed is that for the swap neighborhood, the DM-SD provides bad estimations. We have concluded that this is due to the fact that, for this neighborhood, the attraction basins of the local optima must have a high number of solutions far from it. The methods perform similar for instances of both problems. However, we observe differences in the estimations provided for the different instances considering the same problem. Another important observation derived from this analysis is that the sample size does not have a high influence on the three versions of the DM, while it has to be taken into account if we use the UM. Of course, the higher the sample size considered in the UM, the more accurate the estimations. After observing the statistical analysis, we recommend the following:

- Working under the adjacent swap neighborhood, apply DM-SD
- Considering the swap neighborhood, use DM-ES or DM-SP

We plan to study the estimations provided by these methods considering other different neighborhoods. The performance of the methods, of course, depends on some properties of the landscapes, above all, the distributions of the attraction basins all along the search space. So, we could think of other ways of sampling according to the different distances. This sampling way, should be de-

signed focusing on the specific operator considered, as it is the most important aspect that influences the behavior of these methods. Thus, in order to obtain good results, it is essential to have knowledge beforehand about the properties of the landscape.

Acknowledgements

This work has been supported in part by the Saiotek and Research Groups 20132018 (IT-609-13) Programs (Basque Government) and in part by the Spanish Ministry of Science and Innovation under Grant TIN2013-41272P.

References

1. Albrecht, A., Lane, P., Steinhofel, K.: Combinatorial landscape analysis for k-SAT instances. In: Proceedings of IEEE Congress on Evolutionary Computation. pp. 2498–2504. IEEE Press, Hong Kong (June 2008)
2. Albrecht, A., Lane, P., Steinhofel, K.: Analysis of Local Search Landscapes for k-SAT Instances. *Mathematics in Computer Science* 3(4), 465–488 (2010)
3. Ereemeev, A.V., Reeves, C.R.: Non-parametric Estimation of Properties of Combinatorial Landscapes. In: Cagnoni, S., Gottlieb, J., Hart, E., Middendorf, M., Raidl, G. (eds.) *Applications of Evolutionary Computing, Lecture Notes in Computer Science*, vol. 2279, pp. 31–40. Springer Berlin Heidelberg (2002)
4. Ereemeev, A.V., Reeves, C.R.: On Confidence Intervals for the Number of Local Optima. In: Cagnoni, S., Johnson, C., Cardalda, J., Marchiori, E., Corne, D., Meyer, J.A., Gottlieb, J., Middendorf, M., Guillot, A., Raidl, G., Hart, E. (eds.) *Applications of Evolutionary Computing, Lecture Notes in Computer Science*, vol. 2611, pp. 224–235. Springer Berlin Heidelberg (2003)
5. Grundel, D., Krokhmal, P., Oliveira, C., Pardalos, P.: On the Number of Local Minima for the Multidimensional Assignment Problem. *Journal of Combinatorial Optimization* 13(1), 1–18 (2007)
6. Hernando, L., Mendiburu, A., Lozano, J.A.: An evaluation of methods for estimating the number of local optima in Combinatorial Optimization Problems. *Evolutionary Computation* 21(4), 625–658 (2013)
7. Hernando, L., Pascual, J.A., Mendiburu, A., Lozano, J.A.: A study on the complexity of TSP instances under the 2-exchange neighbor system. In: *Foundations of Computational Intelligence (FOCI), 2011 IEEE Symposium on*. pp. 15–21 (April 2011)
8. Irurozki, E., Calvo, B., Lozano, J.A.: An R package for permutations, Mallows and Generalized Mallows models. *Journal of Statistical Software* (2016)
9. Mattfeld, D.C., Bierwirth, C.: A Search Space Analysis of the Job Shop Scheduling Problem. *Annals of Operations Research* 86(0), 441–453 (1999)
10. Reeves, C.R., Ereemeev, A.V.: Statistical analysis of local search landscapes. *Journal of the Operational Research Society* 55(7), 687–693 (July 2004), <http://dx.doi.org/10.1057/palgrave.jors.2601611>
11. Tayarani-Najaran, M.H., Prügell-Bennett, A.: On the landscape of combinatorial optimization problems. *IEEE Transactions on Evolutionary Computation* 18(3), 420–434 (2014)
12. Tomassini, M., Vérel, S., Ochoa, G.: Complex-network analysis of combinatorial spaces: The NK landscape case. *Physical Review E* 78(6), 66–114 (December 2008)